

MARVisT: Authoring Glyph-based Visualization in Mobile Augmented Reality

Zhutian Chen, Yijia Su, Yifang Wang, Qianwen Wang, Huamin Qu, Yingcai Wu

Abstract—Recent advances in mobile augmented reality (AR) techniques have shed new light on personal visualization for their advantages of fitting visualization within personal routines, situating visualization in a real-world context, and arousing users' interests. However, enabling non-experts to create data visualization in mobile AR environments is challenging given the lack of tools that allow in-situ design while supporting the binding of data to AR content. Most existing AR authoring tools require working on personal computers or manually creating each virtual object and modifying its visual attributes. We systematically study this issue by identifying the specificity of AR glyph-based visualization authoring tool and distill four design considerations. Following these design considerations, we design and implement MARVisT, a mobile authoring tool that leverages information from reality to assist non-experts in addressing relationships between data and virtual glyphs, real objects and virtual glyphs, and real objects and data. With MARVisT, users without visualization expertise can bind data to real-world objects to create expressive AR glyph-based visualizations rapidly and effortlessly, reshaping the representation of the real world with data. We use several examples to demonstrate the expressiveness of MARVisT. A user study with non-experts is also conducted to evaluate the authoring experience of MARVisT.

Index Terms—Personal Visualization, Augmented Reality, Mobile Interactions, Authoring Tool

1 INTRODUCTION

WITH the growing availability of data in our daily life, personal visualization (PV), which aims to help the public better visually explore data of interest in a personal context, has experienced a recent research surge, attracting attention from researchers in the visualization and human-computer interaction (HCI) communities [28]. Unlike traditional visualization applications that are oriented to domain experts, PV is less task focused [28], [48] and is highly correlated to daily life. The growth in mobile augmented reality (AR) techniques (ARKit [3] on iOS and ARCore [24] on Android) have shed new light on PV. We argue that mobile AR techniques have great potential to facilitate PV. First, mobile devices, which have become indispensable in our daily routines, represent one of the most available all-in-one platforms to collect, process, and display data related to personal context. Second, by using the AR techniques, users can situate visualization into a real-world setting, thus enhancing the association between the data being visualized and the real background [15], [20]. However, mainstream desktop-based AR content authoring tools (e.g., Unity and Vuforia) are not suitable for non-experts to create mobile AR visualization. Two gaps primarily contribute to such unsuitability: 1) the steep learning curve of these fully-featured tools, and 2) the inconvenient offline workflow wherein users must work back-and-forth between mobile devices and desktop PCs. These characteristics have motivated us to design and develop a mobile authoring tool for the efficient creation of AR visualization.

This research takes the first step towards enabling non-experts to use mobile devices to create data visualizations in AR environments. We specifically focus on the authoring of glyph-based data visualization for two reasons. First, as a type of infographic, the advantage of glyph-based visualization meets the requirements of PV, namely, conveying data messages quickly and attractively to the public [11], who usually only have limited experience with data visualization. Second, glyph-based visualization objectifies each data point as a physical object that is suitable to display in three-dimensional (3D) AR environments, whereas 3D data charts remain controversial [14], [45]. We abbreviate the term *AR glyph-based visualization* to *ARGVis* for brevity.

Although various existing works presented methods [36], [43], [49], [53] to assist users in creating visualizations, explorations on mobile authoring tools for ARGVis are still scarce. Numerous open questions remain about creating ARGVis in mobile devices. To identify the specificity and challenges of authoring ARGVis, we first compare the abstract workflow of traditional visualization authoring tools with ARGVis authoring tools to assess the similarities and differences between them. Findings reveal that the characteristics of ARGVis are critical in the design of the tools. Thus, we situate ARGVis within the context of the literature to characterize its essential components and the relationships among these components. Based on the characterization, we identify that while traditional authoring tools focus on relationships between data and visual channels, ARGVis authoring tools should further address the relationships between reality and virtuality. Specifically, an ARGVis authoring tool should help users create visualizations that reveal the relationships between 1) data and virtual glyphs, 2) real objects and virtual glyphs, and 3) real objects and data. How to support these kinds of authoring remains an open question. To address this issue, we distill and articulate a set of four design considerations based on our analysis and existing authoring tools, and refine them through several design iterations. Following these design considerations, we present MARVisT, a **Mobile AR**

- Z. Chen is with the State Key Lab of CAD&CG, Zhejiang University and Hong Kong University of Science and Technology. A part of this work was done when Zhutian Chen was a visiting student supervised by Yingcai Wu at the State Key Lab of CAD&CG, Zhejiang University. E-mail: zhutian.chen@connect.ust.hk.
- Y. Su, Y. Wang, Q. Wang and H. Qu are with Hong Kong University of Science and Technology. E-mail: {yijiasu, yifangwang, qwangbb}@connect.ust.hk, huamin@cse.ust.hk.
- Y. Wu is with the State Key Lab of CAD&CG, Zhejiang University. Yingcai Wu is the corresponding author. E-mail: ycwu@zju.edu.cn.

Visualization Tool, tailored for non-experts to create creative visualizations in AR environments. Compared with the state-of-art visualization authoring tools, MARVisT leverages information from reality to assist users in authoring visual mapping, visual scales, and layouts for creating ARGVis. To demonstrate the expressiveness and usefulness of MARVisT, we present examples that are difficult and tedious for non-experts to construct in traditional PCs. We also conduct a user study with participants without visualization and AR expertise to evaluate the authoring experience of MARVisT.

To summarize, our contributions are as follows:

- A systematical study of the specificities of ARGVis authoring tools and the characterization of ARGVis.
- A mobile authoring tool which features context-aware nudging, visual scales synchronization, and virtual glyphs auto-layout to enable non-experts to create ARGVis.
- An evaluation with examples and a user study to demonstrate the expressiveness, usability and usefulness of the tool.

2 RELATED WORKS

2.1 Visualization in Augmented Reality

Recent years have seen a surge in the popularity of consumer-grade AR devices (*e.g.*, head-mounted displays (HMD), mobile phones, and tablets) which has led to an increasing number of studies exploring the visualization and visual analytics based on AR techniques. In the visualization community, the works related to AR can be roughly classified according to the computing modalities that create the AR content, namely, HMD systems [17], [27] and mobile handheld systems [20], [63]. Currently, HMD-based devices have various deficiencies, including narrow field-of-view, low resolution, and inaccurate tracking [9]. Moreover, the HMD devices are still relatively new to most consumers. Bach et al. [6] reported that participants needed additional time to become familiar with the HMD devices; otherwise, they would perform worse in the interactive exploration tasks.

As another type of AR device, mobile devices are more widely available and familiar to the masses. Several works based on mobile AR devices have been introduced for various application scenarios. MelissAR [21], a tablet AR system, showing on-site drift information to assist beekeepers in understanding bee behaviors. Based on real-time face recognition, Zhao et al. [63] developed an iPhone-based ego-centric network AR visualization system supporting on-site analysis of temporal co-authoring patterns of researchers. Nicola et al. [46] proposed a sketch-like AR visualization to support the field workflow of geologists. Despite the extensive use of mobile AR to present data in various professional fields, studies have rarely applied such techniques in casual scenarios, such as people's daily lives (*e.g.*, using mobile AR for shopping assistance [20]), which is one of the most common contexts for using mobile devices. We envision that the personal context is a promising direction for applying mobile AR techniques. Compared with traditional charts, ARGVis can be aesthetically pleasing and easy to understand with the advantage of providing context information directly. Such characteristics help users comprehend and recall information [10], [25]. ARKit [3] and ARCore [24] have been released recently, targeting the iOS and Android platforms, respectively. These infrastructures provide great opportunities to utilize mobile AR techniques for visualization in the personal context. We take the first step toward exploring the application of mobile AR to visualize data for the masses.

2.2 Personal Visualization

PV focuses on visualizing data in a personal context. We mainly focus on the mobile PV systems. Additionally, as some AR content is displayed in the form of real objects, we also survey physical PV systems, which present data using real objects. Mobile PV systems visualize data on mobile devices (*e.g.*, mobile phone and tablet), presenting visualizations to users in a personal context ubiquitously. Researchers have presented mobile PV systems [19], [37], [38] to visualize daily information collected from peripheral sensors. Such mobile systems represent data using standard charts or metaphorical visualizations. The former is tedious, whereas the latter (whose metaphors are defined by the designers) may be obscure for general users. Compared with mobile systems, physical PV systems are innovative in their data presentation forms, which provide abstract data with a concrete physical form [33]. Existing works [32], [34] have investigated physical visualization and provided empirical evidence that it can improve users' efficiency in information retrieval tasks. Some researchers have designed physical visualization to visualize personal data. Khot et al. [35] designed SweatAtoms, a system that transforms heart rate data into five kinds of 3D printed material artifacts. Stusak et al. [56] designed four types of sculptures to visualize running activities and evaluated their usability through a field study. Their findings confirmed that a physical PV could positively change personal and social behaviors. However, the physical form of a visualization is difficult to craft. In our tool, users can create visualizations in the form of physical objects based on AR techniques.

2.3 Visualization Toolkits

Researchers have developed various authoring tools to assist authors in creating visualizations. Compared to programming toolkits [12], [50], [54] and WIMP UI systems (*e.g.*, Lyra [53], iVisDesigner [49], and Data-Driven Guide [36]) for information visualization, pen and touch enable a natural user interface to leverage the human directed manipulation skills, having received attention from visualization researchers in recent years [40], [43], [61], [62]. SketchStory [40] can quickly create data-bound visualizations on a digital white-board for improvisational purposes. Mendez et al. [43] designed and implemented iVoLVER, which could create formal visualizations by adopting an interactive visual language. The most relevant work is DataInk [62], proposed by Xia et al., an authoring tool which supports rigorous creations of data-driven glyph-based visualizations through direct pen and touch input. We adapt DataInk's approaches on touch interaction to the AR scenario. Although these systems investigated methods to create data visualizations with pen and touch input, they mainly focus on the 2D visualization on a 2D canvas, while AR environments involve 3D. On the other hand, while several touch- [26] or pen-based [39], [51] authoring tools have been developed to help users create general AR content, they do not support the creation of data-driven visualization in AR environments. Most recently, Sicat et al. [55] presented DXR, a toolkit which provides both programmatic and graphical user interfaces to help designers create prototypes of immersive visualizations rapidly. However, DXR adopts a form-filling paradigm in the GUI which may not work well in touchscreen devices [61]. Besides, DXR cannot extract the information from reality to facilitate the creations. We draw on this line of works to design a tool that enables non-experts to author data visualization in mobile AR.

3 DESIGN RATIONALE

Given that past work has rarely studied ARGVis, we try to identify the specificity of authoring ARGVis through a comparison between the abstract workflows of traditional visualization authoring tools and ARGVis authoring tools. The differences between these two kinds of tools require us to characterize ARGVis to inform the design of authoring tools. Thus we shape the ARGVis based on the literature and finally propose our design considerations.

3.1 The Specificity of Authoring ARGVis

To identify the specificity and challenges in creating ARGVis, we compare the abstract workflows of two kinds of authoring tools.



Fig. 1. Traditional visualization authoring tools mainly focuses on relationships between data and visual channels.

Traditional Visualization Authoring Tools. One of the core processes of information visualization is mapping data dimensions to visual channels (*e.g.*, size, color, position) [41]. The primary workflow of traditional authoring tools is depicted in Figure 1, wherein the input is data and the output is the visualization.

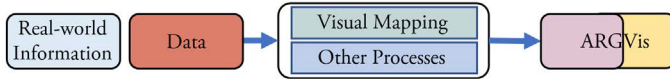


Fig. 2. ARGVis authoring tools have various input and output, requiring additional processes to handle relationships between virtuality and reality.

ARGVis Authoring Tools. Compared to traditional visualization authoring tools, ARGVis authoring tools (shown in Figure 2) have two different features: 1) additional input, which is information about the real-world environment, and 2) a different output, which is ARGVis. These differences raise one critical question for designing an ARGVis authoring tool: What are the characteristics of ARGVis?

3.2 The Characteristics of ARGVis

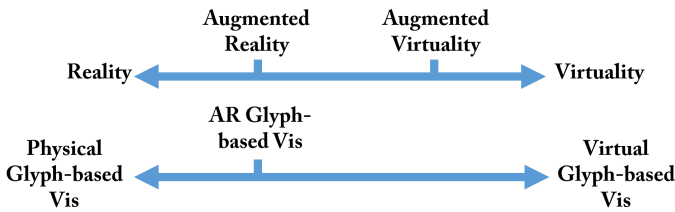


Fig. 3. The glyph-based Vis version of the reality-virtuality continuum.

Sorting out the characteristics of ARGVis can help us identify the specificity of authoring ARGVis and form our design considerations. However, to the best of our knowledge, there is no existing work that investigates ARGVis. Hence, we attempt to characterize ARGVis by considering and summarizing a broad range of previous works in visualization.

We first learn lessons from studies of **AR Visualization**. A few works exist that have investigated visualizing information in AR. Bach et al. [7] proposed AR-CANVAS that depicted the preliminary design space of AR visualization. Willett et al. [59] illustrated *situated* and *embedded visualization*. Although these works focus on general AR visualization, they inform us that ARGVis has two essential components, namely, reality and virtuality.

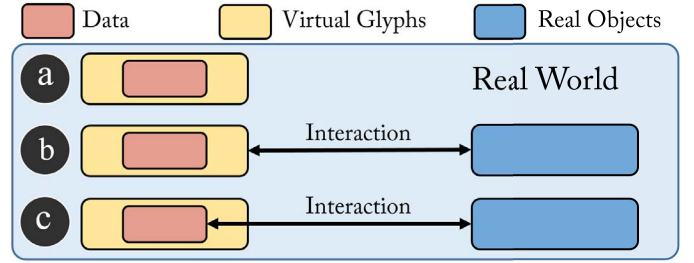


Fig. 4. The relationship between reality and virtuality can be a) weak, b) medium, and c) strong.

To recognize the related works as much as possible, we further situate the ARGVis in the reality-virtuality continuum [44] (Figure 3), which encompasses all possible variations and compositions of reality and virtuality. Focusing on the context of glyph-based visualization, we recognize the left extreme of the continuum, which consists solely of reality, corresponds to **Physicalization** [33], which visualizes data using physical glyphs. Meanwhile the right extreme, which consists solely of virtuality, corresponds to **Glyph-based Visualization** [11], that visualizes data using virtual glyphs.

Besides, we note that although ARGVis is close to and shares commonalities with Physicalization (*e.g.*, using 3D real objects as visual marks and bringing data into the real-world), they have an inherent difference: ARGVis is not entirely tangible, especially the data part, making it more like a physicalization without a tangible body. This characteristic denotes that ARGVis is also relevant to the concept of **Using Concrete Scales** [18] which visualizes data with familiar real objects but does not rely on their tangible body.

By investigating the reality and virtuality, and the relationships between them in AR Visualization, Physicalization, Glyph-based Visualization, and Using Concrete Scales, we conclude two key components of ARGVis as follows:

- **Virtuality** consists of virtual glyphs that represent data points using visual channels to encode data attributes [11], [33];
- **Reality** is the real-world environment including physical objects [7], [59];

We summarize three relationships between them and describe the rationales and challenges in creating ARGVis to reveal these relationships as follows:

- In the **Weak Relationship**, the reality only provides a situation background and does not interact with the virtuality directly (Figure 4a). In this scenario, ARGVis can bring data into the real world [33], and provides a sense of presence [16], increasing users' situational awareness. To create ARGVis in the *weak relationship*, users mainly focus on mapping data attributes to visual channels. However, the AR environment [7] is more complex and dynamic, thereby requiring users to be more careful about visual mappings than in traditional authoring tools.

- In the **Medium Relationship**, real objects, which serve as references, such as the *container* or the *anchor*, are visually related to the 3D virtual glyphs (Figure 4b). Chevalier et al. [18] proposed the concept of *using concrete scales* as the process of visually relating complex measures (*e.g.*, extreme magnitudes or unfamiliar units) with familiar objects from the real world. Their work inspires us to propose the *medium relationship*. Typically, a real object is used as a measure along one of its properties. The real object is interpreted for symbolic meaning when measuring along its non-visible properties or for direct comparison when measuring along its visible properties (*i.e.*, visual channels). When using a visual channel of a real object for comparison, the corresponding channel of virtual glyphs must be on the same scale. Otherwise, the comparison is meaningless.

• In the **Strong Relationship**, real objects are associated with the data (represented by 3D glyphs) as physical referents [7], [59] (Figure 4c). Willett et al. [59] defined the visualizations in this scenario as *embedded visualization*, one characteristic is being the “*positioning individual presentations of data close to their corresponding physical referents.*” In the *strong relationship*, it is tedious for users to manually lay out virtual glyphs close to their physical referents one by one. Furthermore, this error-prone process may lead to problematic visualization when the positions of virtual glyphs are inaccurate.

Our summarization can serve as a profile that highlights the roles of virtual and real parts in ARGVis, directing our attention to design the authoring tool.

3.3 Design Considerations

We adopt an iterative strategy to formulate our design considerations to guide the design. These design considerations, which are rooted in targeted users (non-experts) and the domain (mobile AR), are first initialized based on the pipeline of *InfoVis* [11], [29], [31], relevant authoring tools [49], [53], [62], and our analysis of authoring visualization in mobile AR (Section 3.2), then refined through our experiences across multiple design iterations.

DC1. Balance the Expressivity and Simplicity. An authoring tool should allow users to customize most of the design for expressiveness. Nevertheless, the tool should also lower the barrier for designing as average users usually do not have enough skill or time to fine-tune their designs [28]. This requires the tool to make a tradeoff between the expressivity, which may require comprehensive functionalities, and simplicity, which may reduce the expressivity but lower the barrier for non-experts.

DC2. Aid Visual Mapping with Context Information. In a weak relationship, the real-world canvas requires users to be more careful about visual mapping than in traditional visualization authoring tools [52], [60], because it may bias users’ visual perception of the visualization [58]. It is difficult for the users, especially for those who lack sufficient knowledge of visualization, to manually adjust the mapping considering both the data and the real-world contexts. The tool should leverage context information to assist users in visual mapping to avoid the pitfalls.

DC3. Enable Visual Scale Synchronization between Real Objects and Virtual Glyphs. In a medium relationship, when using a visual channel of a real object for comparison, the corresponding channel of virtual glyphs must be on the same scale.

The tool should help users to efficiently synchronize the visual scales between reality and virtuality for a meaningful comparison.

DC4. Support Auto-layout of Virtual Glyphs. In a strong relationship, it is tedious and error-prone for users to manually place virtual glyphs close to their physical referents one by one. An ARGVis authoring tool should help users map data points with their physical referents and automatically place them close together.

4 MARViT

We present MARViT, a mobile tool that incorporates our design considerations. We first describe the general workflow utilizing a usage scenario. Then, we introduce the design details of each part.

4.1 Usage scenario

Suppose Bob goes to a museum on the last day of his journey. During his visit, an infographic map depicting the path of Napoleon’s march to Moscow attracts Bob’s attention. Bob wants to visualize his travel route in Russia on the map for comparison to get a sense of history. His mobile phone has recorded various travel data, including routes (*e.g.*, distance, and duration of travel), hotels (*e.g.*, cost and rank), and daily expenses (*e.g.*, cost). Bob exports the data into MARViT. In the visual mapping interface (Figure 5a), he chooses the shoe, the house, and the money stack models to represent the route, hotel, and daily expense data, respectively. MARViT automatically duplicates the models for each data point, distributing them methodically on the map.

To encode the *distance* of his routes, he taps on a shoe to open the palette for creating the visual mapping. The palette looks like a double-ring semi-annulus (Figure 5b), where the inner ring consists of beads representing data attributes and the outer ring displays the visual channels. Currently, MARViT supports a suite of frequently-used visual channels, such as *size* (1D-length, 2D-area, 3D-volume), *color*, *angle*, and *opacity*. When Bob taps the *distance* bead, MARViT recommends the *x-width* channels by highlighting the bead based on Bob’s current point-of-view and the ordered data type of *distance*. Following the suggestion, Bob creates the mapping by dragging the *x-width* bead and dropping it to the *distance* bead. He modifies the value of *x-width* so that the shoe model has the same scale as the map. Then, Bob leverages the *y-height* channel of the house models to encode the *cost* attribute. MARViT propagates the same kind of mapping to all data points

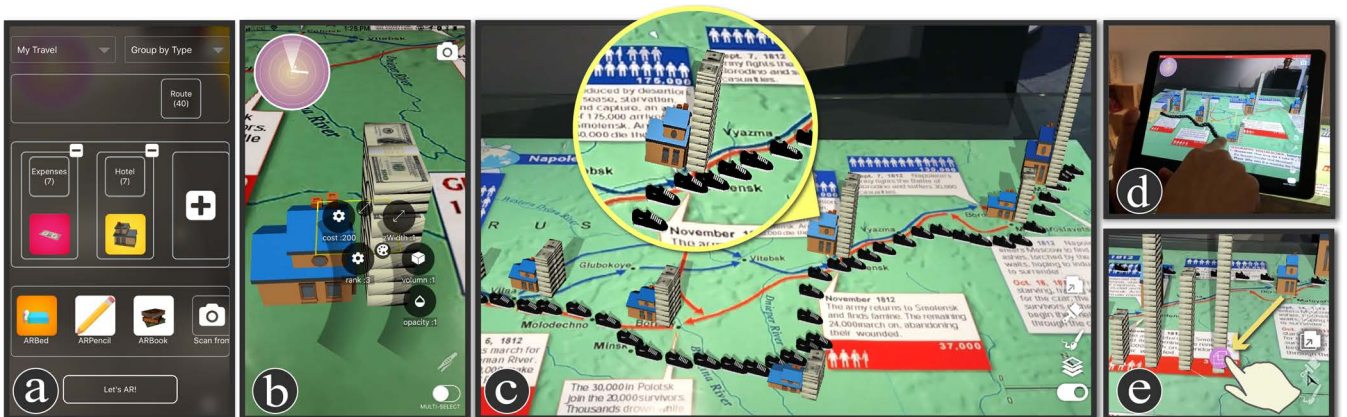


Fig. 5. a) The UI for mapping data points to virtual glyphs. b) The UI for visual mapping. The height of hotels and money stacks represents the cost. The color of houses denotes the rank of hotels. Users can further switch different interaction modes using the buttons in the bottom-right corner. c) The ARGVis blends the designer’s routes (shoes), hotels (houses), and daily expenses (money stacks) data with a real-world infographic map. d) The *Layout Sketch* allows users to lay out numerous objects efficiently. e) The *Copy Layout* method objectifies layouts of collections as small beads.

that have the same attribute. Hence all the money stack models, which also have the *cost* attribute, are likewise encoded with the *y-height* channel (Figure 5b). Next, Bob decides to encode the *rank* attribute of houses using the *z-length* channel. As the *y-height* channel of houses has already been used for visualizing the *cost* attribute, which may lead to interference with the *z-length* channel, MARViT sends a warning message to notify Bob of this pitfall. Thus, he uses the *color* channel to encode the *rank* instead.

MARViT enables users to lay out the glyphs through several methods. By dragging the houses to their cities one-by-one, Bob visualizes the hotel information on the map. Then, Bob groups the data points into three collections based on model types and invokes the *Copy Layout* function to transfer the house layouts to the money stack collection (Figure 5e), thus distributing the money stacks alongside the corresponding houses. Now he can easily compare hotel cost and expenses every day. It is time to portray the itinerary on the map. Although it is possible to recognize a map and place virtual models based on GPS data automatically, the infographic map is customized, leading to difficulty in recognition and unpredictable mismatches between the route data and the map. Therefore, Bob needs to place these virtual glyphs by himself. By using the *Layout Sketch* function, he sketches a path along with the house and money stacks on the screen to distribute the shoes on the map (Figure 5d), revealing his travel route. Comparing his routes to Napoleon's, he discovers that his path is similar to that of Napoleon's retreat from Moscow (Figure 5c). "Uh-huh! Such a fading empire", he muses. Bob captures this ARGVis as an image and sends it to his friends.

4.2 Basic Pipeline

To incorporate DC1, MARViT adopts a *bottom-up strategy* [11], [29] (Figure 6) to guarantee the expressivity and an *objectifying strategy* [61], [62] to ensure the simplicity of interactions.

Map Data Points to Visual Marks. MARViT currently only supports data in tabular format. After importing data, users can drag different groups of data points from the data gallery and 3D glyphs from the glyph gallery to the middle slots (Figure 5a) to specify the mapping between them. Currently, MARViT only supports model fetching from servers, since using advanced methods, such as reconstruction from real objects using a depth camera [30], to acquire 3D models is out of the scope of this research.

Create, Modify, and Remove Visual Mappings. While the form-filling paradigm acts as the core of WIMP UI, such designs are tedious, slow, and error prone in touchscreen devices [61]. Inspired by DataInk [62], we objectify data attributes and visual channels as interactive beads that can be directly manipulated to get rid of form-filling paradigm. Users can create a mapping through a drag-and-drop interaction (Figure 6b). Then, MARViT

will automatically propagate the mapping to the corresponding data attribute and visual channel of all virtual glyphs based on their data values. To modify the scale of a bound visual channel, users can activate the modification state and then slightly drag the bead up or down to alert the value (Figure 6c). The modification will again be propagated to all glyphs. For deletions, users only need to drag the visual channel away from the paired data attribute. The motion will clean the mapping from all the glyphs.

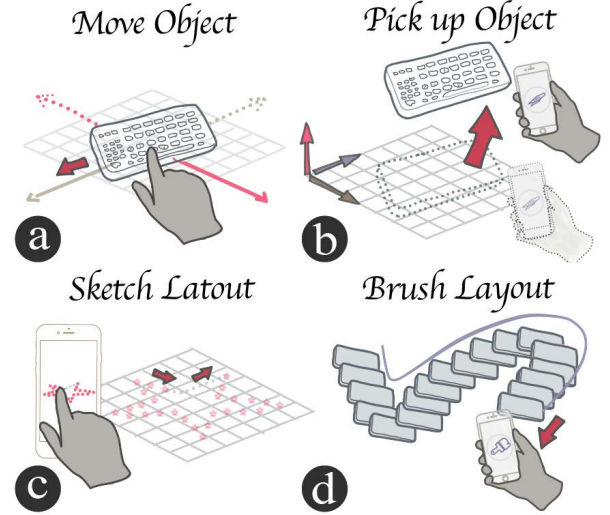


Fig. 7. MARViT provides intuitive interactions that allow users to lay out single or multiple virtual glyphs in both 2D and 3D space efficiently.

Lay Out the Glyphs. After the visual mapping, users can further author the layout of virtual glyphs. MARViT follows the suggestion of previous works [42] to avoid cross-dimensional interactions, supporting four kinds of interactions, namely, $(2D, 3D) \times (Glyph, Collection)$, to lay out the virtual glyphs.

- **2D Interactions on Glyph Level.** MARViT enables users to directly manipulate the position of virtual glyphs using familiar gestures (Figure 7a). For simplification, the object is limited to move on the plan it located when interacting through touching.
- **3D Interactions on Glyph Level.** As for the 3D positions of the glyphs, we design a *Picker* mode (Figure 7b), wherein users can use the mobile as tongs to grip and lift glyphs in the 3D space.
- **2D Interactions on Collection Level.** MARViT allows users to group glyphs into collections based on data attributes. A collection is treated as a single object of a higher level, to which the interactions above can also be applied. Inspired from DataInk [62], we design a *Layout Sketch* method through which users can draw a path on the screen (Figure 7c) to distribute glyphs in the selected collection along the path. This method will only modify the horizontal positions of glyphs.

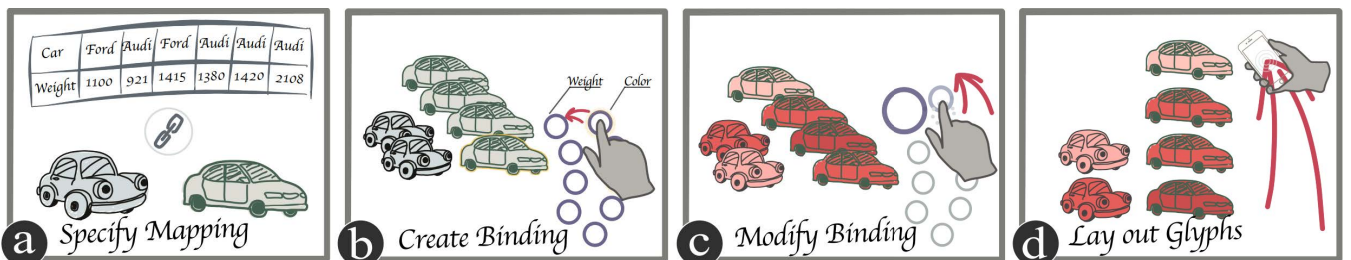


Fig. 6. A pipeline based on a bottom-up strategy [29] to create glyph-based visualizations. a) Specifying the mapping from data points to virtual glyphs. b) Creating visual mappings through direct manipulations. c) Modifying the scale of visual mappings through one hand operations. d) Laying out virtual glyphs through intuitive and effective interactions.

- *3D Interactions on Collection Level.* To modify the 3D position of a collection, users can also use the *Picker* mode to position a collection anywhere in the 3D space. We further design a *Layout Brush* to enable the laying out of glyphs in the 3D space (Figure 7d). Users can wave their mobile like a physical brush in the air to distribute the virtual glyphs in the selected collection to the real-world 3D space.

To facilitate the creation and save users from tedious work, we design several smart functions for efficiently laying out glyphs for data visualization purposes, which can be extended in the future:

- *Side-by-side Comparisons.* By triggering the *Copy Layout* method, users can just drag the layout of the source collection and drop it to the target collection to transfer the desired layout.
- *Stacking Glyphs.* MARVisT will detect the positions of glyphs in a collection whenever its layout is modified and help users to align glyphs automatically for stacking glyphs (Figure 6d).

4.3 Context-aware Nudging

The complex and dynamic AR environment [7] brings difficulties for visual mappings. Considering both best practices guideline drawn from prior research [41], [45], [58] and the context information, MARVisT validates the usage of visual channels before and after creating a visual mapping and guides users with the validation results through an indirect way (DC2).

Validation. From the perspective of perceptual effectiveness, MARVisT considers the following features in the validation to cover all supported visual channels:

- *Orientation.* The size channels (e.g., *1D-length*, *2D-area*, and *3D-volume*) are widely used for ordered data. In the AR environment, the perspective distortion, an important depth cue of the 3D environment, strongly interferes with these size channels [58]. Thus an intuitive assumption is that if a size channel has components along the depth axis, it will be ineffective. MARVisT calculates the depth component of each size channel based on users' perceiving *orientation* and invalidates the one which has the depth component. One limitation of this heuristic rule is that in mobile AR environments the authoring perspective might not be the viewing one. Nevertheless, this rule is still effective as the authoring perspective is usually the best angle for viewing and the user can be informed of the pitfalls of using depth channels.
- *Luminance Contrast.* Whenever users encode data using optical channels (e.g., *color* and *opacity*), MARVisT validates the *luminance contrast* between the virtual glyphs and the real-world background. Adequate *luminance contrast* should be ensured for perceiving shape, form, and other details [58]. MARVisT adopts the root-mean-squared (RMS) method [23] to calculate the contrast ratio given that the AR environment is closer to natural images. A minimum contrast ratio between a pattern and its background is suggested to be 3:1 [58]. MARVisT will fail the validation if the visual encoding cannot meet the minimum contrast ratio. While our rule aims to ensure the legibility, the color issue in the AR environment is far more complex and involves many physical and physiological factors. We further discuss color issues in Section 7 and leave them for future work.
- *Rotational Symmetry.* For the *angle* channels, MARVisT will validate the *rotational symmetry* of the 3D glyph. MARVisT will fail the validation if the 3D glyph's order of *rotational symmetry* about the axis is greater than four (means that the rotation by 90° does not change the object) since the perceptual accuracy of angles drops off within the exact horizontal and vertical [58].

- *Separability.* MARVisT also considers that the visual channels have already been used to avoid subsumption and perceptual integration [58]. For example, when encoding data attribute a_i , if the *x-width* channel is used for attribute a_j , *area* channels will be invalidated to avoid subsumption; *y-height* channel will also be invalidated to maintain the separability of the channels.

TABLE 1
Permitted Visual Channels of Data Types.

Data Types	Visual Channels*
Quantitative	length (x-, y-, z-) > angle (φ -, θ -, ψ -) > area (top-, left-, front-) > color (luminance or saturation) > volume
Ordinal	color (luminance or saturation) > length (x-, y-, z-) > angle (φ -, θ -, ψ -) > area (top-, left-, front-) > volume
Nominal	color (hue)

* Ordered by perceptual effectiveness rankings.

Nudging. Some existing works [60] automatically generate the visualization encoded with the most effective visual channels and prohibit any ineffective combinations of them. However, targeting the masses produces the requirements of enabling flexible and expressive creations for greater engagement, which may violate design principles of visualization. Considering these subtle requirements of general users, we prefer to nudge users instead of prohibiting them with hard constraints. Nudging is a concept that proposes positive reinforcement and indirect suggestions to influence users [57]. MARVisT integrates a twofold nudging mechanism, meaning it, nudges users in 1) what to do and 2) what not to do. Specifically, when the user is ready to encode a data attribute with visual channels, MARVisT will generate a list of valid visual channels ordered by perceptual effectiveness [41], [45], [58] in relation to the data type (Table 1). It would then recommend the most effective one to the user through a jumping animation of the bead. When the user finishes creating or modifying a visual encoding, MARVisT will validate the visual encoding and warn the user through messages if the validation fails.

Admittedly, our initial design is modest. Although more advanced context-aware nudging methods are possible, the primary goal of this proof-of-concept work is to explore and design features of an overall approach to enable non-experts to create ARGVis. In place of more advanced methods, we intentionally use interpretable, deterministic heuristics for nudging users with context awareness. We envision the design and evaluation of improved context-aware nudging systems as an important future research direction.

4.4 Visual Scales Synchronization

As described in DC3, if a visual channel of real objects (referred to as *RVC*) is used for comparison, the corresponding visual channel of virtual glyphs (referred to as *VVC*) must be on the same scale as the *RVC*. For example, if a designer wants to compare the *volume* channel of Earth, Neptune, and Jupiter using a ping pong ball as a benchmark for familiarity and intuition, he needs to change the size of any planet to the same size as the ball (e.g., the Earth in Figure 8) and scale other planets accordingly to maintain the relative ratio of their *volume* (converting their *volume* channels from the original scale to the ping pong ball scale).

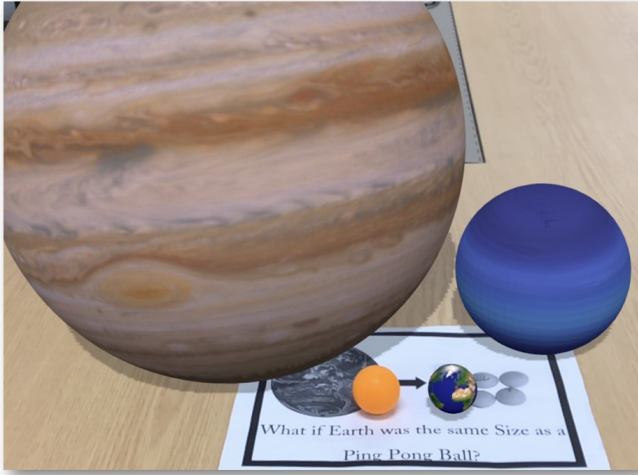


Fig. 8. Users can extract the *volume* of the ping pong ball and assign it to the Earth. Other virtual planets (Neptunus and Jupiter) will be scaled accordingly if the corresponding visual mapping on the *volume* exists.

Although the user can manually adjust a *VVC* to synchronize with an *RVC* via the basic interactions (Section 4.2), it is tedious and ineffective. We design a semi-automatic function based on object detection techniques [3], [5] to improve the effectiveness. After detecting a real object, MARVisT automatically extracts the values of its visual channels, such as *size*, *position*, *text*, and so on. Limited by the computer vision technique, currently, MARVisT only supports an approximation of these values. For instance, we estimate a real object's size channels based on its bounding box. Also, MARVisT can only detect text on a pure color background.

When the user taps on this real object, a single-ring semi-annulus similar to the one in Figure 5b will pop out. The semi-annulus consists of beads which represent the available visual channels of the real object. With a similar interaction, the user can drag a bead and drop it onto a virtual glyph to assign its value to the corresponding visual channel of the virtual glyph. If this visual channel has already been used to encode a data attribute, MARVisT will inversely calculate the new scale based on the value of the *RVC* and the data bounded with the virtual glyph. Then the new scale will automatically be propagated to all virtual glyphs with the same visual mapping, leading to updates of corresponding *VVCs*. For more technical details on the object detection, the visual channel extraction, and Figure 8, we refer the reader to the appendix.

4.5 Virtual Glyphs Auto-layout

As demonstrated in DC4, an authoring tool should help users to automatically place virtual glyphs (referred to as *VGs*) together with their corresponding real objects (referred to as *ROs*). For example, without supporting the automatic laying out process, a user needs to manually move the three virtual sugar stacks one-by-one to create what we see in Figure 9.

For the auto-layout, the tool must first detect *ROs*, map data points to the *ROs*, and then place *VGs* of the data points to the positions of their corresponding physical referents. For real object detection, we can reuse the methods mentioned in section 4.4. As for the second step, to enable the one-to-one mapping process fully automatically without using a hard-code method is non-trivial, given the identifiers of the *ROs* depending on the underlying recognition algorithm. For instance, the milk in Figure 9 can be recognized as different identifiers (e.g., *milk* or *drink*, etc.) using different computer vision libraries, leading to a difficulty in



Fig. 9. The sugar content of drinks. If there are n virtual sugar stacks needed to be placed, the user needs to perform $O(n)$ interactions. With the auto-layout, the user can finish the task using $O(1)$ interactions.

mapping the data points to the *ROs*. To decouple the identification of *ROs* from the recognition algorithms and design a generalizable approach for mapping data points to *ROs*, we propose to use a semi-automatic approach that asks the designer to specify the mapping by using the data attributes of data points and the visual channels of *ROs*. We chose visual channels to depict a real object because 1) the visual channel is one of the most accessible features for a mobile phone with a camera, 2) the values of visual channels are independent of the underlying recognition algorithm, and 3) it is convenient to reuse visual channels that have already been extracted in Section 4.4. Moreover, our approach does not require extra UI widgets and interactions. Specifically, to specify the mapping, the user can drag the bead of a visual channel of the *ROs* and drop it onto the bead of a data attribute of the *VGs*. MARVisT will map the *ROs* to the data points by their index in the selected channel and attribute. For example, the user can drag the bead of the *volume* channel of the drinks in Figure 9 and drop it onto the *sugar content* attribute of the data points (presented by sugar stacks) to map them by index: the data point with the highest *sugar content* to the drink with the highest *volume*, and so on. Finally, MARVisT will automatically place each glyph to the position of its corresponding real object. The user can further fine-tune the *VGs*' positions through the layout interactions introduced in Section 4.2.

Currently, MARVisT only supports mapping between categorical data and ordinal data with monotonic relationships. Other cases are left for future works. The technical details and the step-by-step explanations of creating Figure 9 are enclosed in the appendix.

5 IMPLEMENTATION AND PERFORMANCE

MARVisT is a universal mobile application implemented on iOS with techniques that include React Native [22], ARKit [3], Vision [5], and SceneKit [4]. The tracking of real-world environments and registration of virtual glyphs are supported by ARKit, a powerful infrastructure that enables the development of AR applications. Also, in each frame of the ARKit session, we utilize the HSL image of the field-of-view to calculate the luminance contrast ratio and the environment light estimation to improve the realism. To recognize real objects and estimate visual channel values from them, MARVisT leverages both the Vision (wherein the model for object recognition was trained in Turi Create using Darknet YOLO) and ARKit. We embedded a toy dataset of reference objects in MARVisT to supplement and improve the performance of extracting visual channel values. The dataset can then be extended and shifted to the cloud in the future. MARVisT

adopts a responsive design that allows adaptation to mobile devices with different screen sizes. The design of MARVisT does not depend on unique techniques listed above.

We conducted experiments to evaluate the construction times, the frame rates of the static scenes and the dynamic scenes for varying data sizes running on an iPhone 8 Plus and an iPad Pro to identify the limits of MARVisT. The detailed experiments' settings and results are found in Appendix B. Overall, for a dataset with reasonable size (1, 000 models or less), our implementation achieves a low construction time (in around 2 seconds) and real-time frame rates (over 50 FPS in most cases). Considering the usage scenario (i.e., the personal context) of MARVisT, wherein users likely do not have big datasets to visualize, we think the performance of the current implementation is acceptable. We believe MARVisT still can be improved in the future, such as optimizing the memory usage and following the best practice, *etc.*

6 EVALUATION

MARVisT was evaluated from two perspectives. First, we presented a gallery of works created by MARVisT to illustrate its expressiveness. Then an in-lab user study with non-experts was conducted to evaluate MARVisT's usability and usefulness.

6.1 Example Personal Visualizations

To demonstrate its expressiveness, we created a diverse set of ARGVis using MARVisT. MARVisT supports a wide range of glyph-based visualizations from one glyph encoded with a single visual channel to a collection of glyphs with multiple visual channels. Moreover, devising engaging real-world infographics using MARVisT is more efficient than manually designing physical

visualizations, including material selection and object placement, which illustrators and designers are usually responsible for.

MARVisT provides several visual channels for encoding data, which is essential for attaining data-driven design. For example, Figure 10a presents a sample visualization that combines virtual glyphs with real-world content thereby depicting previously-used mobile phones of the designer whose sizes visually encode the associated price. Users can also encode various data attributes with different visual channels (Figure 10c), or use multiple channels to encode a data attribute for multiple encoding (Figure 10b, e).

Besides, customizations become increasingly flexible when users manipulate the layout of virtual glyphs in collections. For example, a user can distribute the glyphs along a sketch path on a plane to create a customized layout (Figure 10d). Figure 10e organizes the glyphs in a calendar layout. MARVisT also supports the laying out of glyphs in 3D space through the *Layout Brush* methods and enables users to stack glyphs with minimal effort by guessing the users' brush intentions (10f).

Utilizing the object detection technique, MARVisT can reduce users' workload in creating designs. Similar to the case in Figure 8, MARVisT can extract the geometric properties (e.g., top-view size) of real objects and assign them to the bars in Figure 10b, efficiently adjusting their size to make them fit with the real background key-caps. Additionally, through building connections between visual features of real objects (e.g., the *text* of the whiteboard in Figure 10a, key-caps in Figure 10b, and the calendar in Figure 10e) and data attributes of virtual glyphs, MARVisT can automatically place multiple virtual glyphs onto their corresponding physical referents, thereby improving the efficiency.

Adopting the data-driven strategy, MARVisT dynamically updates the graphics according to changes of the dataset. For example, if the price of mobile phones in Figure 10a is changed, the

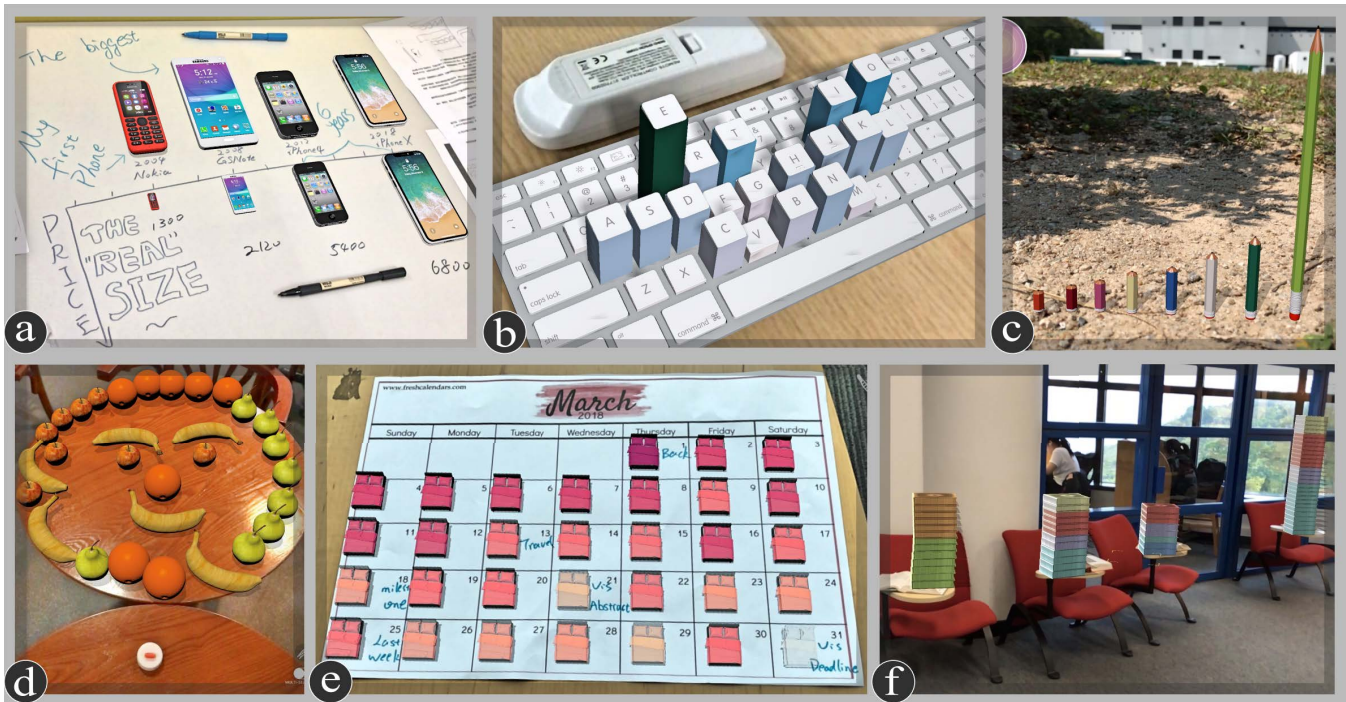


Fig. 10. Examples created with MARVisT. a) An isotype chart combines the reality (the sketching on the whiteboard) and the virtual glyphs (the eight mobile phones). The four mobile phones in the second row visualize their price using the size channel. b) Michael Knuepfel's keyboard frequency sculpture recreated and enhanced using MARVisT. Both the color and height channels are used for double encoding. c) Moodley brand identity's physical infographics recreated by MARVisT using color and height to encode various data attributes. d) A smiling face created using the *Layout Sketch* method. The face consists of fruits the designer eat in a week. e) A calendar visualization created using the auto-layout method. The design displays the sleep duration per night in the March 2018 using the color and opacity for double encoding. f) A stacked glyph chart created using the *Layout Brush* and the automatically stacking function. It depicts the designer's favorite seat in the library. Color represents the day of the week.

size of mobiles will be updated accordingly. Given the data-driven feature and the physical form of virtual glyphs, it is highly efficient to devise physical infographics with MARVisT. For instance, we recreated the physical infographic Figure 10b from Michael Knuepfel [1] and Figure 10c from moodley brand identity [2]. A designer using MARVisT does not need to purchase, craft, and place the physical materials but can still obtain the same visual effects as the traditional physical infographics does.

6.2 User Study

Next, we conducted a laboratory study to examine if non-experts could create ARGVis using MARVisT and if there were any usability issues for improving the system. The study also aimed to gain of presenting data in the form of ARGVis.

6.2.1 Participants and Experimental Setup

We recruited 14 participants (5 males; ages: 21 to 27, average: 23.6) from the university through email announcements and billboard advertisements. We pre-screened participants to ensure that they all had no or limited experience in data visualization design. According to the pre-study survey, participants had diverse major backgrounds, including Computer Science (4), Chemistry (2), Electronic Engineering (2), Economics (1), Social Science (1), Food Science and Engineering (1), Plant Protection (1), Digital Media (1), and Big Data Technology (1). They were also not the co-authors of the paper. All of them reported having prior experience in mobile AR, but none had created AR content before. The participants were identified as P1-P14, respectively. Each participant received a gift card worth \$6.37 at the beginning of the session, independent of their performance. The study was run in a conference room which was $5m \times 4m$, using a 10.5-inch iPad Pro.

6.2.2 Procedure

Participants accomplished a consent form and a demographics questionnaire, completed three tasks with MARVisT, and then concluded with a post-study survey and an interview.

Demonstration and Training (25min). The investigator demonstrated the concept of glyph-based visualization in mobile AR and then introduced MARVisT, including its functions and interactions in the glyph mapping, visual mapping, and layout steps through several walk-through examples. Participants were asked to exactly replicate the provided examples, and seek guidance and assistance whenever necessary. The participants were also encouraged to freely try and explore every function and interaction until they felt confident about using the tool.

Replication (25min). The investigator proceeded to the replication task. In this phase, the participants were asked to replicate two data visualizations (Figure 5c and Figure 10b, referred to as RT1 and RT2) for two new datasets based on the real-world background (the infographic map and the keyboard). When the task began, participants received documentation about the datasets and several color images with legends of the targeted visualizations. We designed the visualizations to be sufficiently rich in features to enable participants to reach most of the functions in MARVisT during the replication. The study setup was informal, allowing participants to interrupt at any time to ask questions.

Freeform Exploration (20 min). Next, the investigator asked participants to freely explore the tool by designing their own data visualizations with pre-loaded datasets.

Post-study Survey and Interview (20 min). Participants were asked to complete a post-study survey using a 7-point Likert scale

(1-strongly disagree, 7-strongly agree). The questions assessed the usability (Q1-Q4) and expressiveness (Q5) for DC1, the usefulness (Q6-Q8) of the features to accommodate DC2-DC3, and the engagement (Q9) using MARVisT. The investigator then conducted a semi-structured interview to collect qualitative comments.

6.2.3 Results

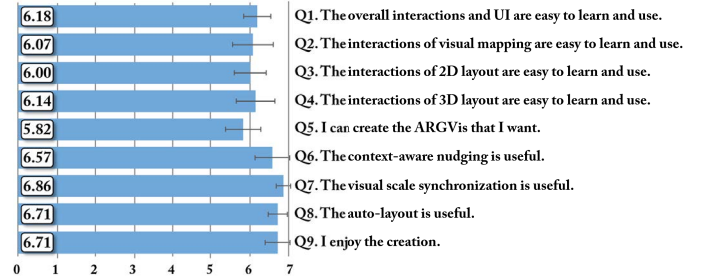


Fig. 11. All questions received an averaging rating greater than five, which was very encouraging.

All the participants completed the replication tasks with minimal guidance in roughly 13 min in total. The third task was open-ended and involved interview-like conversations between participants and the investigator that derived useful feedback for the tool and ARGVis. Figure 11 presents the questions and average user ratings. Our analysis of subjective ratings and qualitative comments of the participants led to insights into the usability, expressiveness, usefulness, and engagement of the authoring experience.

Usability. Targeted at general users, MARVisT must be both easy to learn and use. Overall, the usability of MARVisT was appreciated by the participants based on the result of Q1 ($\mu = 6.18$, 95% $CI = [5.83, 6.53]$). Specifically, based on the results of Q2, Q3, and Q4, the participants responded positively about the the main interactions of MARVisT (all the ratings of the three questions were not lower than 6). They found the objectification UI design to be “*effective and reasonable*” (P1, P3, P4, P6-P14) and rated the interactions for visual mapping as easy to learn and use ($\mu = 6.07$, 95% $CI = [5.55, 6.59]$). The participants also felt it was easy to learn and use the 2D layout interactions ($\mu = 6.00$, 95% $CI = [5.59, 6.41]$) and the metaphorical 3D layout interactions ($\mu = 6.14$, 95% $CI = [5.64, 6.64]$). During the study, the participants were all curious about how to move the glyphs in the *height* dimension when we asked them to do so. After showing our solutions, most of the participants were impressed with our approach for laying out objects in 3D space because “*it is much better than dragging the three axes on (the) 2D screen to move objects in 3D space.*” (P11). They found these layout methods not only to be intuitive and efficient but also “*engaging*” (P3) and “*fun to use*” (P4).

Expressiveness. Compared with the usability, there is still room for improvement when it comes to the expressiveness with regards to the participants’ rating in Q5, “I can create the ARGVis that I want” ($\mu = 5.82$, 95% $CI = [5.37, 6.27]$). Seven participants pointed out that the inability of text input makes MARVisT fail to allow adding graphics guides (*e.g.*, title, axis, legend, etc.). We proposed to add voice input methods in the future to support text input while still avoiding keyboard input. All the participants were satisfied with our proposal. Another criticism about the expressiveness is the lack of ability to create virtual glyphs, which is beyond the scope of this study and has been left for the future. Nevertheless, we observed that none of the participants created the same designs though they were unembellished given the constraints of time, data, and the real-world environment (*e.g.*, Figure 12).



Fig. 12. “Apples and Apple” designed by a participant in the third task, depicting the money to buy an Apple phone can buy dozens of apples. One apple in the graphics represents a dozen apples.

Usefulness. When considering the AR features that leverage information from reality to facilitate the authoring, the participants commented positively and confirmed their usefulness (Q6 - Q8).

The participants agreed that the context-aware nudging mechanism, which can improve the usability, was useful (Q6, $\mu = 6.57$, 95% $CI = [6.12, 7.02]$). Most of the participants gave positive feedback, and commented that “*it provides stepping stones to help users initialize the design.*” (P13) and “*it is like a grammar checker, helping me to avoid errors.*” (P5) We observed that only three participants pointed out the perspective distortion problem when encoding data with the size channels that have the most components on the depth axis. This indicated our consideration of providing design critiques based on both real-world background and design principles is reasonable and necessary for non-experts.

As indicated by Q7, the participants spoke highly of how we utilize the information from reality to facilitate authoring visual scales, and rated that the visual scales synchronization was useful ($\mu = 6.86$, 95% $CI = [6.67, 7.05]$). For this feature, the participants favored it because it not only “*can dramatically improve the efficiency*” (P6) but also “*is impressive as I have never seen this (feature) before.*” (P9) The intuitiveness of this feature is another reason the participants appreciated it, e.g., “*this feature is self-explanatory and the results (of interaction) are as expected.*” (P9)

In contrast, although the participants also favored the auto-layout method, and rated that it was useful in Q8 ($\mu = 6.71$, 95% $CI = [6.46, 6.96]$), they found it was not as intuitive as the visual scales synchronization feature. P11 noted, “*this feature is useful especially when there are multiple objects to place, but it took me some time to understand how it works.*” Comments indicated this was caused by the participants’ lack of familiarity with the concept of creating mappings between objects through selecting attributes (JOIN), e.g., “*...I am not a CS guy. I have never used this before.*” (P10) However, we observed that once the participants got the idea of how it works, they could quickly use it for rapidly placing glyphs with their physical referents. Several participants expressed that the cost of time to learn this feature was acceptable for the benefit it brought, e.g., “*Every time you want to use a powerful tool you need to learn something.*” (P14)

Engagement. The participants also responded positively on visualizing information in AR form. The rating on the statement of Q9 that “*I enjoy the creation.*” ($\mu = 6.71$, 95% $CI = [6.39, 7.03]$) suggested that MARViST aroused their creative desires. All the participants agreed that MARViST worked in the right direction

and AR techniques had great potential in presenting information, especially in the personal context: “*I would like to use traditional charts when reporting data in a work context, but AR infographics is a better choice for personal context.*” (P13)

Suggestions for Improvements. Additionally, the participants also provided constructive comments that imply further improvements. A more powerful and intelligent template for initialization was wanted because “*sometimes it is hard to decide where to start.*” (P5) For the limitation of creating virtual glyphs, P8 suggested that MARViST can allow users to create approximate glyphs using the bounding box and texture of real objects as “*sometimes users do not need high fidelity 3D reconstruction models.*” We also observed that the participants like to place virtual glyphs in different places to see how they interact with real objects, inspiring us to transfer functions like SnapToReality [47] to further improve the efficiency. Another shortcoming that affected users’ experiences is the performance of AR techniques, including the accuracy of tracking and registration. Other minor issues such as missing Un-Do functions are ignored as we do not aim to develop a full-featured system.

7 DISCUSSION

7.1 Mobile AR for Personal Visualization

AR techniques are attracting considerable attention from the field of visualization because of the popularity of low-cost AR devices, especially mobile AR devices, which are one of the most accessible devices in our daily life. In the wake of this surge of AR techniques, we aim to explore the possibility of utilizing these methods in data visualization. However, this potential technique is the topic of a heated debate. On the one hand, AR visualization offers two inherent benefits, namely, the engaged representation, which stands out from existing media, and its context-preserving capability, which seamlessly blends the digital information with the real-world context into a cohesive view. On the other hand, visualizations in AR environments are criticized for the occlusion and distortion issues caused by their 3D form and rendering space, which lead to inaccuracies in presenting data and errors in analytical tasks.

We make the observation that AR techniques are well suited for personal contexts, in which accuracy is not the first priority [28], and other aspects, such as aesthetics and engagement, play important roles [13]. Utilizing AR techniques in the personal context can fully leverage the advantages of AR for creative and engaging presentations and circumvent the disadvantages of 3D visualizations that are not suitable for analytic purposes. Moreover, as accessible all-in-one platforms, mobile devices bridge the gap between AR techniques and general users, bringing opportunities to make use of AR for PV practical.

Based on this observation, we took the first step toward the use of mobile AR techniques in the personal context by designing and developing MARViST, which enables non-experts to use mobile devices to present their data together with the context that inspires their creations. Although the infographics created by MARViST are visually embellished by real-object glyphs, which can have a negative impact on certain analytic tasks, it is now generally understood that embellishment is not equivalent to chart junk [8], [36]. Visual embellishments can help present and remember the context of data [10] that makes it suitable for PV [25]. The feedback from the user study reflected participants’ preferences of ARGVis for presentations and communication in the personal context.

We envision that AR techniques will bring considerable potential to PV and mobile devices are more accessible than HMD

devices for general users in the near future. Thus, we hope that our work blazes a trail and arouses interest in applying mobile AR techniques for PV. Although MARVisT is an authoring tool targeted at non-experts, the three identified relationships between reality and virtuality and the four proposed design considerations can be generalized, and guide the design of advanced tools for other users. Moreover, the interaction we designed can be easily adapted to other mobile AR applications with various purposes.

7.2 AR Visualization Authoring Tool

Designing an AR visualization authoring tool relies on research from several fields. Essentially, it highly depends on the research on visualizing information in AR. Visualizing information in AR and authoring AR visualization have different challenges: the former focuses on the challenges [7] like *How to avoid visual clutter and occlusion?*; *What visualization types to choose?*; *How to maintain faithful perception?* and the like; while the latter considers *expressiveness*, *efficiency*, and *usability*, and so on. Nevertheless, research on visualization in AR can guide and foster the design of AR visualization authoring tools. Without further investigation into AR visualization, many design decisions of MARVisT remain unclear. For example, when encoding data using the color channel, which color we should use – the material color or the screen color – is still ambiguous. Generally, in traditional visualization, we always use the screen color to encode data. However, in AR environments, the screen color of a virtual object is determined by several factors, such as the object’s material, and the lighting of the environment. For high fidelity, we should encode data using material color, but it may result in a contrast effect [58] that may bias users’ perception. Will this issue affect users’ perception? How can we address it? Besides, the answers to questions about creation pipeline and visualization types in AR determine the *expressiveness* of the authoring tool. These open questions require further study.

In addition, the design of authoring tools is influenced by progress in other research fields. The *Computer Vision* techniques can improve the *efficiency* of the tool and the connection between virtuality and reality. For example, advanced *Object Detection* techniques can enhance the visual scales synchronization and the auto-layout features. With the advancement of *3D Scanning* techniques which allow the extraction of high quality and structural information from the reality, MARVisT could support users to reconstruct 3D glyphs from real objects rather than fetch them from remote servers. On the other hand, the *usability* of the tool mainly depends on the user interfaces and interactions on mobile devices, which is an important topic in the HCI field. MARVisT adapts UI designs of DataInk [62], an authoring tool which supports data-driven sketching with pen and touch, to support fluent interactions on mobile devices. We believe a good authoring tool can encourage more attempts at AR visualization, which in turn fosters collecting requirements and designing better AR authoring tools. With this work, we hope to encourage researchers to pay attention to AR visualization, eventually revealing the potential of this novel field.

7.3 Limitations and Future Work

The user study demonstrates that the participants were able to create ARGVis with MARVisT in a short time and were satisfied with MARVisT subjectively. However, these results should be treated carefully as well as qualitative studies with small sample size. The participants also cannot fully represent the general public. Further evaluations are warranted to confirm the results can be applied more

generally. Also, the results do not prove the interactions and UI designs of MARVisT are optimal but only indicate that the overall designs have high usability. We did not compare MARVisT with other tools since currently, to the best of our knowledge, we cannot find a similar tool as the benchmark. Moreover, the laboratory environment is not entirely equal to the personal context. A field study is required to evaluate MARVisT in a real personal context.

Most of the limitations proposed by the users during the study were related to tool maturity. For example, a text input function is wanted; and functions which allow creating customized 3D models are missing. MARVisT also involves certain inherent limitations, which we consider as opportunities for future research. First, users have to manually place the virtual glyphs if they want to encode data attributes using position channels, or to place virtual glyphs to their physical referents without creating the one-to-one mappings between them. MARVisT should better support data binding on position channels to prevent incorrect positioning. Second, MARVisT currently does not support certain types of visualizations, such as node-link graphs and charts with axes. Designing methods that support these visualizations in mobile devices with limited screen size need further work. Third, although the primary disadvantages caused by the 3D are not as critical in PV as in visualizations serving analytics purposes, these problems still trouble users when perceiving designs. Fourth, the context-aware nudging mechanism is built upon heuristic rules and is not flexible enough. We left the more powerful mechanism, which may not only be built upon heuristic rules, for future research.

8 CONCLUSION

In this work, we present MARVisT, a mobile AR glyph-based visualization authoring tool tailored for non-experts. MARVisT aims at utilizing the mobile AR techniques to tackle several unique challenges of PV by enabling non-experts to use mobile devices for creating glyph-based visualizations in AR environments. Our design of MARVisT is informed by existing principles and pipelines for authoring visualizations, and a characterization of ARGVis that help to identify the specificity of ARGVis authoring tools. Through MARVisT, users can bind the visual channels to data attributes and lay out virtual glyphs in 3D space through direct manipulation and intuitive interactions. MARVisT also leverages information from reality to assist non-experts in visual mapping, visual scales synchronization, and auto-layout. We demonstrate the expressiveness, usability, and usefulness of MARVisT through a gallery of examples and a user study.

ACKNOWLEDGMENTS

The work was supported by National Key R&D Program of China (2018YFB1004300), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1609217), NSFC (61761136020, 61502416), Zhejiang Provincial Natural Science Foundation (LR18F020001) and the 100 Talents Program of Zhejiang University. This project was also partially funded by RGC (CRF 16241916), Microsoft Research Asia, and the research fund of the Ministry of Education of China (188170-170160502). The authors would like to thank Maps.com, the owner of the infographics Napoleon’s Russian Campaigns, as well as the anonymous reviewers for their valuable comments.

REFERENCES

- [1] Michael Knuepfel. <https://vimeo.com/20277306>.
- [2] moodley brand identity. <https://moodley.at/caritas-streetwork-publishing>.
- [3] Apple. ARKit. <https://developer.apple.com/arkit/>.
- [4] Apple. SceneKit. <https://developer.apple.com/scenekit/>.
- [5] Apple. Vision. <https://developer.apple.com/documentation/vision>.
- [6] B. Bach, R. Sicat, J. Beyer, M. Cordeil, and H. Pfister. The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? *IEEE TVCG*, 24(1):457–467, 2018.
- [7] B. Bach, R. Sicat, H. Pfister, and A. Quigley. Drawing into the AR-Canvas: Designing Embedded Visualizations for Augmented Reality. *Workshop on Immersive Analytics of IEEE VIS*, 2017.
- [8] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. In *Proc. of CHI*, pp. 2573–2582. ACM, 2010.
- [9] J. Baumeister, S. Y. Ssin, N. A. M. ElSayed, J. Dorrian, D. P. Webb, J. A. Walsh, T. M. Simon, A. Irlitti, R. T. Smith, M. Kohler, and B. H. Thomas. Cognitive Cost of Using Augmented Reality Displays. *IEEE TVCG*, 23(11):2378–2388, 2017.
- [10] R. Borgo, A. Abdul-Rahman, F. Mohamed, P. W. Grant, I. Reppa, L. Floridi, and M. Chen. An empirical study on using visual embellishments in visualization. *IEEE TVCG*, (12):2759–2768, 2012.
- [11] R. Borgo, J. Kehr, D. H. S. Chung, E. Maguire, R. S. Laramée, H. Hauser, M. Ward, and M. Chen. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Proc. of Eurographics STARs*, pp. 39–63. The Eurographics Association, 2013.
- [12] M. Bostock, V. Ogievetsky, and J. Heer. D³ Data-Driven Documents. *IEEE TVCG*, 17(12):2301–2309, 2011.
- [13] J. Boy, F. Détienne, and J. Fekete. Storytelling in Information Visualizations: Does it Engage Users to Explore Data? In *Proc. of CHI*, pp. 1449–1458. ACM, 2015.
- [14] R. Brath. 3D InfoVis is Here to Stay: Deal with It. *Workshop on 3DVis of IEEE VIS*, 2014.
- [15] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstück, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive Analytics. In *Proc. of BDVA*, pp. 73–80. IEEE, 2015.
- [16] M. Chen, K. Gaither, N. W. John, and B. McCann. Cost-benefit Analysis of Visualization in Virtual Environments. *CoRR*, abs/1802.09012, 2018.
- [17] Z. Chen, H. Qu, and Y. Wu. Immersive Urban Analytics through Exploded Views. *Workshop on Immersive Analytics of IEEE VIS*, 2017.
- [18] F. Chevalier, R. Vuillemot, and G. Gali. Using Concrete Scales: A Practical Framework for Effective Visual Depiction of Complex Measures. *IEEE TVCG*, 19(12):2426–2435, 2013.
- [19] M. Chiu, S. Chang, Y. Chang, H. Chu, C. C. Chen, F. Hsiao, and J. Ko. Playful Bottle: A Mobile Social Persuasion System to Motivate Healthy Water Intake. In *Proc. of Ubicomp*, pp. 185–194. ACM, 2009.
- [20] N. A. M. ElSayed, B. H. Thomas, K. Marriott, J. Piantadosi, and R. T. Smith. Situated Analytics. In *Proc. of BDVA*, pp. 96–103. IEEE, 2015.
- [21] U. Engelke, H. Hutson, H. Nguyen, and P. A. de Souza. MelissAR: Towards Augmented Visual Analytics of Honey Bee Behaviour. In *Proc. of CHI*. ACM, 2016.
- [22] Facebook. React Native. <https://facebook.github.io/react-native/>.
- [23] R. A. Frazor and W. S. Geisler. Local Luminance and Contrast in Natural Images. *Elsevier Vision Research*, 46(10):1585–1598, 2006.
- [24] Google. ARCore. <https://developers.google.com/ar/>.
- [25] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proc. of CHI*, pp. 1191–1200. ACM, 2015.
- [26] A. Henrysson, M. Ollila, and M. Billinghurst. Mobile Phone Based AR Scene Assembly. In *Proc. of MUM*, pp. 95–102. ACM, 2005.
- [27] D. Herr, J. Reinhardt, R. Krueger, G. Reina, and T. Ertl. Immersive Visual Analytics for Modular Factory Layout Planning. *Workshop on Immersive Analytics of IEEE VIS*, 2017.
- [28] D. Huang, M. Tory, B. A. Aseniero, L. Bartram, S. Bateman, S. Carpendale, A. Tang, and R. Woodbury. Personal Visualization and Personal Visual Analytics. *IEEE TVCG*, 21(3):420–433, 2015.
- [29] S. Huron, Y. Jansen, and S. Carpendale. Constructing Visual Representations: Investigating the Use of Tangible Tokens. *IEEE TVCG*, 20(12):2102–2111, 2014.
- [30] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. W. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of UIST*, pp. 559–568. ACM, 2011.
- [31] Y. Jansen and P. Dragicevic. An Interaction Model for Visualizations Beyond The Desktop. *IEEE TVCG*, 19(12):2396–2405, 2013.
- [32] Y. Jansen, P. Dragicevic, and J. Fekete. Evaluating the Efficiency of Physical Visualizations. In *Proc. of CHI*, pp. 2593–2602. ACM, 2013.
- [33] Y. Jansen, P. Dragicevic, P. Isenberg, J. Alexander, A. Karnik, J. Kildal, S. Subramanian, and K. Hornbæk. Opportunities and Challenges for Data Physicalization. In *Proc. of CHI*, pp. 3227–3236. ACM, 2015.
- [34] Y. Jansen and K. Hornbæk. A Psychophysical Investigation of Size as a Physical Variable. *IEEE TVCG*, 22(1):479–488, 2016.
- [35] R. A. Khot, L. Hjorth, and F. F. Mueller. Understanding Physical Activity through 3D Printed Material Artifacts. In *Proc. of CHI*, pp. 3835–3844. ACM, 2014.
- [36] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-Driven Guides: Supporting Expressive Design for Information Graphics. *IEEE TVCG*, 23(1):491–500, 2017.
- [37] S. Kim and E. Paulos. inAir: Sharing Indoor Air Quality Measurements and Visualizations. In *Proc. of CHI*, pp. 1861–1870. ACM, 2010.
- [38] J. Kjeldskov, M. B. Skov, J. Paay, and R. Pathmanathan. Using Mobile Phones to Support Sustainability: A Field Study of Residential Electricity Consumption. In *Proc. of CHI*, pp. 2347–2356. ACM, 2012.
- [39] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg. Sketching Up the World: In Situ Authoring for Mobile Augmented Reality. *Springer Personal and Ubiquitous Computing*, 16(6):623–630, 2012.
- [40] B. Lee, R. H. Kazi, and G. Smith. SketchStory: Telling More Engaging Stories with Data through Freeform Sketching. *IEEE TVCG*, 19(12):2416–2425, 2013.
- [41] J. D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM TOG*, 5(2):110–141, 1986.
- [42] D. Mendes, F. Caputo, A. Giachetti, A. Ferreira, and J. Jorge. A Survey on 3D Virtual Object Manipulation: From the Desktop to Immersive Virtual Environments. In *CGF*. Wiley Online Library, 2018.
- [43] G. G. Méndez, M. A. Nacenta, and S. Vandenheste. iVoLVER: Interactive Visual Language for Visualization Extraction and Reconstruction. In *Proc. of CHI*, pp. 4073–4085. ACM, 2016.
- [44] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A Class of Displays on the Reality-Virtuality Continuum, 1995.
- [45] T. Munzner. *Visualization Analysis and Design*. A K Peters, 2014.
- [46] J. T. Nagel, N. Gazcon, J. Larregui, M. Ganuza, E. Bjerg, and S. Castro. Immersive Analytics for Geology: Field Sketch-Based Visualization to Assist Geological Structure Analysis During Fieldwork. *Workshop on Immersive Analytics of IEEE VIS*, 2017.
- [47] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson. SnapToReality: Aligning Augmented Reality to the Real World. In *Proc. of CHI*, pp. 1233–1244. ACM, 2016.
- [48] Z. Pousman, J. T. Stasko, and M. Mateas. Casual Information Visualization: Depictions of Data in Everyday Life. *IEEE TVCG*, 13(6):1145–1152, 2007.
- [49] D. Ren, T. Höllerer, and X. Yuan. iVisDesigner: Expressive Interactive Design of Information Visualizations. *IEEE TVCG*, 20(12):2092–2101, 2014.
- [50] P. D. Ritsos, J. Jackson, and J. C. Roberts. Web-based Immersive Analytics in Handheld Augmented Reality. *Poster of IEEE VIS*, 2017.
- [51] D. Saakes, H. Yeo, S. Noh, G. Han, and W. Woo. Mirror Mirror: An On-Body T-shirt Design System. In *Proc. of CHI*, pp. 6058–6063. ACM, 2016.
- [52] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration. *IEEE TVCG*, 23(1):331–340, 2017.
- [53] A. Satyanarayan and J. Heer. Lyra: An Interactive Visualization Design Environment. *Wiley CGF*, 33(3):351–360, 2014.
- [54] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. *IEEE TVCG*, 23(1):341–350, 2017.
- [55] R. Sicat, J. Li, J. Choi, M. Cordeil, W. Jeong, B. Bach, and H. Pfister. DXR: A Toolkit for Building Immersive Data Visualizations. *IEEE TVCG*, pp. 1–1, 2018.
- [56] S. Stusak, A. Tabard, F. Sauka, R. A. Khot, and A. Butz. Activity Sculptures: Exploring the Impact of Physical Visualizations on Running Activity. *IEEE TVCG*, 20(12):2201–2210, 2014.
- [57] R. H. Thaler and C. R. Sunstein. *Nudge: Improving Decisions About Health, Wealth, and Happiness*. Penguin Books, 2009.
- [58] C. Ware. *Information Visualization: Perception for Design*. Elsevier, 2012.
- [59] W. Willett, Y. Jansen, and P. Dragicevic. Embedded Data Representations. *IEEE TVCG*, 23(1):461–470, 2017.
- [60] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE TVCG*, 22(1):649–658, 2016.

- [61] H. Xia, B. Araújo, T. Grossman, and D. J. Wigdor. Object-Oriented Drawing. In *Proc. of CHI*, pp. 4610–4621. ACM, 2016.
- [62] H. Xia, N. H. Riche, F. Chevalier, B. de Araujo, and D. Wigdor. DataInk: Direct and Creative Data-Oriented Drawing. In *Proc. of CHI*. ACM, 2018.
- [63] M. Zhao, Y. Su, J. Zhao, S. Chen, and H. Qu. Mobile Situated Analytics of Ego-centric Network Data. In *Proc. of SIGGRAPH ASIA Symposium on Visualization*, pp. 14:1–14:8. ACM, 2017.



Zhutian Chen is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include AR visualization, immersive analytics, and visual analytics. He received the bachelor's degree from the Department of Software Engineering, South China University of Technology in 2015. For more information, please visit <http://chenzhutian.org>.



Yijia Su is currently a Research Assistant in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He received his B.Eng degree in Software Engineering from South China University of Technology in 2016. His research interests include data visualisation and augmented reality techniques. For more information, please visit <http://www.yijiasu.me/>



Yifang Wang received her BEng degree in software engineering from Zhejiang University, China, in 2018. She will pursue her Ph.D. degree in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST) from Sept. 2018. Her research interests include information visualization, visual analytics, and human computer interaction.



Qianwen Wang is now a Ph.D. student at Hong Kong University of Science and Technology. Her research interest focus on information visualization and interactive machine learning. She received a BS degree in Electronics Engineering, Xi'an Jiaotong University.



Huamin Qu is a professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His main research interests are in visualization and human-computer interaction, with focuses on urban informatics, social network analysis, e-learning, text visualization, and explainable artificial intelligence. He obtained a BS in Mathematics from Xi'an Jiaotong University, China, an MS and a PhD in Computer Science from the Stony Brook University. For more information,

please visit <http://www.huamin.org>.



Yingcai Wu is a ZJU100 Young Professor at the State Key Lab of CAD&CG, Zhejiang University. His main research interests are in information visualization and visual analytics, with focuses on immersive visualization, sports science, social media analysis, and urban computing. He received his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology. Prior to his current position, Dr. Wu was a postdoctoral researcher in the University of California, Davis from 2010 to 2012, and a

researcher in Microsoft Research Asia from 2012 to 2015. For more information, please visit <http://www.ycwu.org>.